N 5

```
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFFFFFFFFFFFFFF         111              111         XXX         XXX
FFF                  111111           111111         XXX         XXX
FFF                  111111           111111         XXX         XXX
FFF                  111111           111111         XXX         XXX
FFF                     111              111            XXX   XXX
FFF                     111              111            XXX   XXX
FFF                     111              111            XXX   XXX
FFFFFFFFFFFF            111              111               XXX
FFFFFFFFFFFF            111              111               XXX
FFFFFFFFFFFF            111              111               XXX
FFF                     111              111            XXX   XXX
FFF                     111              111            XXX   XXX
FFF                     111              111            XXX   XXX
FFF                     111              111         XXX         XXX
FFF                     111              111         XXX         XXX
FFF                     111              111         XXX         XXX
FFF                  111111111        111111111      XXX         XXX
FFF                  111111111        111111111      XXX         XXX
FFF                  111111111        111111111      XXX         XXX
```

_$25

Symb

IOC1
IO_
IO_
IO_
IO_
IO_
KILL

KILL
KILL
LB_
LB_
LB_
LB_
LB_
LOCA
LOCA
LOCK

LOCK
LOCK
LOCK
LOC_
LOC_
L_CO
L_CO
L_DA
L_DA
MAIN
MAKE
MAKE
MAKE
MAKE
MAKE
MAKE

MAKE
MAKE
MAP_
MAP_

MAP
MARK
MARK
MARK
MARK
MARK

FILEID**MAKNMB

```
MM      MM    AAAAAA    KK      KK  NN      NN  MM      MM  BBBBBBBB
MM      MM    AAAAAA    KK      KK  NN      NN  MM      MM  BBBBBBBB
MMMM  MMMM  AA      AA  KK      KK  NN      NN  MMMM  MMMM  BB      BB
MMMM  MMMM  AA      AA  KK      KK  NN      NN  MMMM  MMMM  BB      BB
MM  MM  MM  AA      AA  KK    KK    NNNN    NN  MM  MM  MM  BB      BB
MM  MM  MM  AA      AA  KK    KK    NNNN    NN  MM  MM  MM  BB      BB
MM      MM  AA      AA  KKKKKK      NN  NN  NN  MM      MM  BBBBBBBB
MM      MM  AA      AA  KKKKKK      NN  NN  NN  MM      MM  BBBBBBBB
MM      MM  AAAAAAAAAA  KK    KK    NN    NNNN  MM      MM  BB      BB
MM      MM  AAAAAAAAAA  KK    KK    NN    NNNN  MM      MM  BB      BB
MM      MM  AA      AA  KK      KK  NN      NN  MM      MM  BB      BB    ....
MM      MM  AA      AA  KK      KK  NN      NN  MM      MM  BB      BB    ....
MM      MM  AA      AA  KK      KK  NN      NN  MM      MM  BBBBBBBB      ....
MM      MM  AA      AA  KK      KK  NN      NN  MM      MM  BBBBBBBB      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;  1     0001  0 MODULE MAKNMB (
;  2     0002  0                 LANGUAGE (BLISS32),
;  3     0003  0                 IDENT = 'V04-000'
;  4     0004  0                 ) =
;  5     0005  1 BEGIN
;  6     0006  1
;  7     0007  1 !
;  8     0008  1 !**********************************************************************
;  9     0009  1 !*                                                                    *
; 10     0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                          *
; 11     0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.           *
; 12     0012  1 !*  ALL RIGHTS RESERVED.                                             *
; 13     0013  1 !*                                                                    *
; 14     0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
; 15     0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
; 16     0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
; 17     0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
; 18     0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
; 19     0019  1 !*  TRANSFERRED.                                                       *
; 20     0020  1 !*                                                                    *
; 21     0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
; 22     0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
; 23     0023  1 !*  CORPORATION.                                                       *
; 24     0024  1 !*                                                                    *
; 25     0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
; 26     0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.            *
; 27     0027  1 !*                                                                    *
; 28     0028  1 !*                                                                    *
; 29     0029  1 !**********************************************************************
; 30     0030  1
; 31     0031  1 !++
; 32     0032  1 !
; 33     0033  1 ! FACILITY:  F11ACP Structure Level 1
; 34     0034  1 !
; 35     0035  1 ! ABSTRACT:
; 36     0036  1 !
; 37     0037  1 !       This routine converts a file name string into the
; 38     0038  1 !       RAD-50 name block format.
; 39     0039  1 !
; 40     0040  1 ! ENVIRONMENT:
; 41     0041  1 !
; 42     0042  1 !       STARLET operating system, including privileged system services
; 43     0043  1 !       and internal exec routines.
; 44     0044  1 !
; 45     0045  1 !--
; 46     0046  1 !
; 47     0047  1 !
; 48     0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  2-Jan-1977  17:06
; 49     0049  1 !
; 50     0050  1 ! MODIFIED BY:
; 51     0051  1 !
; 52     0052  1 !       V03-004 CDS0003         Christian D. Saether    2-Jan-1984
; 53     0053  1 !               Use longword addressing on external FIL$ routine.
; 54     0054  1 !
; 55     0055  1 !       V03-003 CDS0002         Christian D. Saether    6-Dec-1983
; 56     0056  1 !               Change LIB$ references to FIL$.
; 57     0057  1 !
```

```
;    58          0058  1 !       V03-002 ACG0302         Andrew C. Goldstein,    3-Dec-1982  13:55
;    59          0059  1 !               Add $ and _ to file names, allow long names
;    60          0060  1 !
;    61          0061  1 !       V03-001 CDS0001         C. Saether                1-Jul-1982
;    62          0062  1 !               Don't force absolute addressing mode when declaring
;    63          0063  1 !               external routine lib$cvt_dtb.
;    64          0064  1 !
;    65          0065  1 !       A0101   ACG0057         Andrew C. Goldstein,    10-Aug-1979  16:41
;    66          0066  1 !               Wild card interface changes
;    67          0067  1 !
;    68          0068  1 !       A0100   ACG00001        Andrew C. Goldstein,  10-Oct-1978  20:03
;    69          0069  1 !       Previous revision history moved to F11A.REV
;    70          0070  1 !**
;    71          0071  1
;    72          0072  1
;    73          0073  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    74          0074  1 REQUIRE 'SRC$:FCPDEF.B32';
;    75          1065  1
;    76          1066  1
;    77          1067  1 ! Linkages to subroutines in this module.
;    78          1068  1 !
;    79          1069  1
;    80          1070  1 LINKAGE
;    81          1071  1         L_GETCHAR       = JSB :
;    82          1072  1                           NOPRESERVE (5)
;    83          1073  1                           GLOBAL (COUNT = 6, STRINGP = 7, FCOUNT = 8),
;    84          1074  1
;    85          1075  1         L_GETSTAR       = JSB :
;    86          1076  1                           GLOBAL (COUNT = 6, STRINGP = 7),
;    87          1077  1
;    88          1078  1         L_TYPE          = JSB :
;    89          1079  1                           GLOBAL (COUNT = 6, STRINGP = 7);
;    90          1080  1
;    91          1081  1 ! Routines in this module
;    92          1082  1 !
;    93          1083  1
;    94          1084  1 FORWARD ROUTINE
;    95          1085  1         MAKE_NAMEBLOCK  : NOVALUE,      ! main routine
;    96          1086  1         GETCHAR         : L_GETCHAR,    ! get RAD-50 set character
;    97          1087  1         GETSTAR         : L_GETSTAR,    ! get star character, if any
;    98          1088  1         TYPE            : L_TYPE;       ! determine type of current character
```

```
;  100          1089  1 GLOBAL ROUTINE MAKE_NAMEBLOCK (LENGTH, STRING, NAMEBLOCK) : NOVALUE =
;  101          1090  1
;  102          1091  1 !++
;  103          1092  1 !
;  104          1093  1 ! FUNCTIONAL DESCRIPTION:
;  105          1094  1 !
;  106          1095  1 !       This routine converts a file name string into the
;  107          1096  1 !       RAD-50 name block format.
;  108          1097  1 !
;  109          1098  1 ! CALLING SEQUENCE:
;  110          1099  1 !       MAKE_NAMEBLOCK (ARG1, ARG2, ARG3)
;  111          1100  1 !
;  112          1101  1 ! INPUT PARAMETERS:
;  113          1102  1 !       ARG1: length of file name string
;  114          1103  1 !       ARG2: address of file name string
;  115          1104  1 !
;  116          1105  1 ! IMPLICIT INPUTS:
;  117          1106  1 !       NONE
;  118          1107  1 !
;  119          1108  1 ! OUTPUT PARAMETERS:
;  120          1109  1 !       ARG3: address of file name block
;  121          1110  1 !
;  122          1111  1 ! IMPLICIT OUTPUTS:
;  123          1112  1 !       NONE
;  124          1113  1 !
;  125          1114  1 ! ROUTINE VALUE:
;  126          1115  1 !       NONE
;  127          1116  1 !
;  128          1117  1 ! SIDE EFFECTS:
;  129          1118  1 !       NONE
;  130          1119  1 !
;  131          1120  1 !--
;  132          1121  1
;  133          1122  2 BEGIN
;  134          1123  2
;  135          1124  2 MAP
;  136          1125  2         NAMEBLOCK       : REF BBLOCK;   ! name block arg
;  137          1126  2
;  138          1127  2 GLOBAL REGISTER
;  139          1128  2         COUNT   = 6,                    ! characters remaining in string
;  140          1129  2         STRINGP = 7     : REF VECTOR [,BYTE],  ! string pointer
;  141          1130  2         FCOUNT  = 8;                    ! count of chars in current field
;  142          1131  2
;  143          1132  2 LOCAL
;  144          1133  2         VERSION,                        ! file version number
;  145          1134  2         P,                              ! string scan pointer
;  146          1135  2         BLOCKP          : REF VECTOR [,WORD]; ! pointer into name block
;  147          1136  2
;  148          1137  2 EXTERNAL ROUTINE
;  149          1138  2         FIL$CVT_DTB     : ADDRESSING_MODE (GENERAL); ! decimal to binary convert
;  150          1139  2
;  151          1140  2 ! Initialize all the locals.
;  152          1141  2 !
;  153          1142  2
;  154          1143  2 CH$FILL (0, NMB$C_LENGTH, .NAMEBLOCK);  ! zero the entire block
;  155          1144  2 STRINGP = .STRING;                      ! set up string pointer
;  156          1145  2 P = CH$FIND_CH (.LENGTH, .STRINGP, ' '); ! look for a terminating space
```

```
;  157          1146  2 COUNT = .P - .STRINGP;                        ! compute count
;  158          1147  2 IF CH$FAIL (.P)
;  159          1148  2 THEN COUNT = .LENGTH;                        ! use whole string if no space
;  160          1149  2 BLOCKP = NAMEBLOCK[NMB$W_NAME];              ! point to name field in block
;  161          1150  2 FCOUNT = 0;                                  ! init chars in field count
;  162          1151  2
;  163          1152  2
;  164          1153  2 ! Build the name field, consisting of 3 words of 3 RAD-50 characters per word.
;  165          1154  2 !
;  166          1155  2
;  167          1156  2 DECR I FROM 3 TO 1 DO
;  168          1157  2     BEGIN
;  169          1158  2     DECR J FROM 3 TO 1 DO
;  170          1159  2         BLOCKP[0] = .BLOCKP[0] * 40 + GETCHAR ();
;  171          1160  2     BLOCKP = .BLOCKP + 2;
;  172          1161  2     END;
;  173          1162  2
;  174          1163  2 ! Eat remaining trailing name field characters.
;  175          1164  2 !
;  176          1165  2
;  177          1166  2 WHILE 1 DO
;  178          1167  2     BEGIN
;  179          1168  2     CASE TYPE () FROM 0 TO 6 OF
;  180          1169  3         SET
;  181          1170  3         [0,4,5,6]: EXITLOOP;
;  182          1171  3         [INRANGE, OUTRANGE]:
;  183          1172  4             BEGIN
;  184          1173  4             COUNT = .COUNT - 1;
;  185          1174  4             STRINGP = .STRINGP + 1;
;  186          1175  4             END;
;  187          1176  3         TES;
;  188          1177  2     END;
;  189          1178  2
;  190          1179  2 IF GETSTAR ()                                 ! set wild card bits if star
;  191          1180  2 THEN
;  192          1181  3     BEGIN
;  193          1182  3     NAMEBLOCK[NMB$V_WILD] = 1;
;  194          1183  3     NAMEBLOCK[NMB$V_ALLNAM] = 1;
;  195          1184  2     END;
;  196          1185  2
;  197          1186  2 ! Pick up the name delimiter, which is either dot or end of string.
;  198          1187  2 !
;  199          1188  2
;  200          1189  2 CASE TYPE () FROM 1 TO 5 OF
;  201          1190  2     SET
;  202          1191  2     [1,2,3,4]:  ERR_EXIT (SS$_BADFILENAME);
;  203          1192  3     [5]:        BEGIN
;  204          1193  3                 COUNT = .COUNT - 1;      ! pick up the character
;  205          1194  3                 STRINGP = .STRINGP + 1;
;  206          1195  3                 END;
;  207          1196  2     [OUTRANGE]: 0;
;  208          1197  2     TES;
;  209          1198  2
;  210          1199  2 ! Now build the type field, consisting of 1 word of 3 RAD-50 characters.
;  211          1200  2 !
;  212          1201  2
;  213          1202  2 FCOUNT = 0;                                  ! re-init chars in field count
```

```
;  214     1203  2
;  215     1204  2 DECR J FROM 3 TO 1 DO
;  216     1205  2     BLOCKP[0] = .BLOCKP[0] * 40 + GETCHAR ();
;  217     1206  2
;  218     1207  2 ! Eat remaining trailing type field characters.
;  219     1208  2 !
;  220     1209  2
;  221     1210  2 WHILE 1 DO
;  222     1211  2     BEGIN
;  223     1212  2     CASE TYPE () FROM 0 TO 6 OF
;  224     1213  2         SET
;  225     1214  2         [0,4,5,6]: EXITLOOP;
;  226     1215  2         [INRANGE, OUTRANGE]:
;  227     1216  2             BEGIN
;  228     1217  4             COUNT = .COUNT - 1;
;  229     1218  4             STRINGP = .STRINGP + 1;
;  230     1219  4             END;
;  231     1220  2         TES;
;  232     1221  2     END;
;  233     1222  2
;  234     1223  2 IF GETSTAR ()                          ! set wild card bits if star
;  235     1224  2 THEN
;  236     1225  2     BEGIN
;  237     1226  2     NAMEBLOCK[NMB$V_WILD] = 1;
;  238     1227  2     NAMEBLOCK[NMB$V_ALLTYP] = 1;
;  239     1228  2     END;
;  240     1229  2
;  241     1230  2 ! Pick up the type delimiter, which may be dot, semicolon, or end of string.
;  242     1231  2 !
;  243     1232  2
;  244     1233  2 CASE TYPE () FROM 1 TO 6 OF
;  245     1234  2     SET
;  246     1235  2     [1,2,3,4]:  ERR_EXIT (SS$_BADFILENAME);
;  247     1236  2     [5,6]:      BEGIN
;  248     1237  2                 COUNT = .COUNT - 1;     ! pick up the character
;  249     1238  2                 STRINGP = .STRINGP + 1;
;  250     1239  2                 END;
;  251     1240  2     [OUTRANGE]: 0;
;  252     1241  2     TES;
;  253     1242  2
;  254     1243  2 ! If the version is not wild card and there are still characters present,
;  255     1244  2 ! get the binary version number.
;  256     1245  2 !
;  257     1246  2
;  258     1247  2 IF GETSTAR ()                          ! set wild card bits if star
;  259     1248  2 THEN
;  260     1249  2     BEGIN
;  261     1250  2     NAMEBLOCK[NMB$V_WILD] = 1;
;  262     1251  2     NAMEBLOCK[NMB$V_ALLVER] = 1;
;  263     1252  2     END
;  264     1253  2 ELSE IF .COUNT GTR 0
;  265     1254  2 THEN
;  266     1255  2     BEGIN
;  267     1256  2     BLOCKP = .BLOCKP + 2;
;  268     1257  3     IF NOT FIL$CVT_DTB (.COUNT, .STRINGP, VERSION)
;  269     1258  3     THEN ERR_EXIT (SS$_BADFILENAME);
;  270     1259  3     IF .VERSION GTRU 32767
```

```
;  271          1260  3      THEN ERR_EXIT (SS$_BADFILEVER);
;  272          1261  3      (.BLOCKP)<0,16> = .VERSION;
;  273          1262  2      END;
;  274          1263  2
;  275          1264  2  RETURN 1;
;  276          1265  2
;  277          1266  1  END;                                   ! end of routine MAKE_NAMEBLOCK

                                                        .TITLE  MAKNMB
                                                        .IDENT  \V04-000\

                                                        .EXTRN  FIL$CVT_DTB

                                                        .PSECT  $CODE$,NOWRT,2

                                            0FFC 00000          .ENTRY  MAKE_NAMEBLOCK, Save R2,R3,R4,R5,R6,R7,R8,- ; 1089
                                                                        R9,R10,R11
                                    5B      0000V CF  9E 00002          MOVAB   GETSTAR, R11
                                    5A      0000V CF  9E 00007          MOVAB   TYPE, R10
                                    5E            04  C2 0000C          SUBL2   #4, SP
                28                  00            6E  2C 0000F          MOVC5   #0, (SP), #0, #40, @NAMEBLOCK           ; 1143
                                                  0C  BC 00014
                                    57      08    AC  D0 00016          MOVL    STRING, STRINGP                         ; 1144
                                67          04    AC  20 3A 0001A          LOCC    #32, LENGTH, (STRINGP)                  ; 1145
                                                  02  12 0001F          BNEQ    1$
                                                  51  D4 00021          CLRL    R1
                                56                51  C3 00023 1$:       SUBL3   STRINGP, P, COUNT                       ; 1146
                                                  51  D5 00027          TSTL    P                                       ; 1147
                                                  04  12 00029          BNEQ    2$
                                    56      04    AC  D0 0002B          MOVL    LENGTH, COUNT                           ; 1148
                            52      0C    AC  06  C1 0002F 2$:       ADDL3   #6, NAMEBLOCK, BLOCKP                   ; 1149
                                                  58  D4 00034          CLRL    FCOUNT                                  ; 1150
                                    59            03  D0 00036          MOVL    #3, I                                   ; 1156
                                    53            03  D0 00039 3$:       MOVL    #3, J                                   ; 1158
                                    54            62  3C 0003C 4$:       MOVZWL  (BLOCKP), R4                            ; 1159
                                    54            28  C4 0003F          MULL2   #40, R4
                                                0000V 30  00042          BSBW    GETCHAR
                                62                54  A1 00045          ADDW3   R0, R4, (BLOCKP)
                                    F0            53  F5 00049          SOBGTR  J, 4$
                                    52            02  C0 0004C          ADDL2   #2, BLOCKP                              ; 1160
                                    E7            59  F5 0004F          SOBGTR  I, 3$                                   ; 1156
                                                  6A  16 00052 5$:       JSB     TYPE                                    ; 1168
                                06                50  CF 00054          CASEL   R0, #0, #6
                        000E    000E            0014    00058 6$:       .WORD   8$-6$,-
                        0014    0014            0014    00060                   7$-6$,-
                                                                                7$-6$,-
                                                                                7$-6$,-
                                                                                8$-6$,-
                                                                                8$-6$,-
                                                                                8$-6$
                                                  56  D7 00066 7$:       DECL    COUNT                                   ; 1173
                                                  57  D6 00068          INCL    STRINGP                                 ; 1174
                                                  E6  11 0006A          BRB     5$                                      ; 1168
                                                  6B  16 0006C 8$:       JSB     GETSTAR                                 ; 1179
                                    10            50  E9 0006E          BLBC    R0, 9$
                                    50      0C    AC  D0 00071          MOVL    NAMEBLOCK, R0                           ; 1182
```

```
                  11    A0          01  88 00075          BISB2   #1, 17(R0)                     ;
                        50     0C   AC  D0 00079          MOVL    NAMEBLOCK, R0                  ; 1183
                  10    A0          20  88 0007D          BISB2   #32, 16(R0)                    ;
                                    6A  16 00081 9$:      JSB     TYPE                           ; 1189
             04         01          50  CF 00083          CASEL   R0, #1, #4                     ;
0098         0098       0098      0098     00087 10$:     .WORD   23$-10$,-                      ;
                                  000C     0008F                  23$-10$,-                      ;
                                                                  23$-10$,-                      ;
                                                                  23$-10$,-                      ;
                                                                  11$-10$                        ;
                                    04  11 00091          BRB     12$                            ;
                                    56  D7 00093 11$:     DECL    COUNT                          ; 1193
                                    57  D6 00095          INCL    STRINGP                        ; 1194
                                    58  D4 00097 12$:     CLRL    FCOUNT                         ; 1202
                        53          03  D0 00099          MOVL    #3, J                          ; 1204
                        54          62  3C 0009C 13$:     MOVZWL  (BLOCKP), R4                   ; 1205
                        54          28  C4 0009F          MULL2   #40, R4                        ;
                                 0000V  30 000A2          BSBW    GETCHAR                        ;
             62         54          50  A1 000A5          ADDW3   R0, R4, (BLOCKP)               ;
                        F0          53  F5 000A9          SOBGTR  J, 13$                         ;
                                    6A  16 000AC 14$:     JSB     TYPE                           ; 1212
             06         00          50  CF 000AE          CASEL   R0, #0, #6                     ;
000E         000E       000E      0014     000B2 15$:     .WORD   17$-15$,-                      ;
             0014       0014      0014     000BA                  16$-15$,-                      ;
                                                                  16$-15$,-                      ;
                                                                  16$-15$,-                      ;
                                                                  17$-15$,-                      ;
                                                                  17$-15$,-                      ;
                                                                  17$-15$                        ;
                                    56  D7 000C0 16$:     DECL    COUNT                          ; 1217
                                    57  D6 000C2          INCL    STRINGP                        ; 1218
                                    E6  11 000C4          BRB     14$                            ; 1212
                                    6B  16 000C6 17$:     JSB     GETSTAR                        ; 1223
                        10          50  E9 000C8          BLBC    R0, 18$                        ;
                        50     0C   AC  D0 000CB          MOVL    NAMEBLOCK, R0                  ; 1226
                  11    A0          01  88 000CF          BISB2   #1, 17(R0)                     ;
                        50     0C   AC  D0 000D3          MOVL    NAMEBLOCK, R0                  ; 1227
                  10    A0          10  88 000D7          BISB2   #16, 16(R0)                    ;
                                    6A  16 000DB 18$:     JSB     TYPE                           ; 1233
             05         01          50  CF 000DD          CASEL   R0, #1, #5                     ;
003E         003E       003E      003E     000E1 19$:     .WORD   23$-19$,-                      ;
                        000E      000E     000E9                  23$-19$,-                      ;
                                                                  23$-19$,-                      ;
                                                                  23$-19$,-                      ;
                                                                  20$-19$,-                      ;
                                                                  20$-19$                        ;
                                    04  11 000ED          BRB     21$                            ;
                                    56  D7 000EF 20$:     DECL    COUNT                          ; 1237
                                    57  D6 000F1          INCL    STRINGP                        ; 1238
                                    6B  16 000F3 21$:     JSB     GETSTAR                        ; 1247
                        11          50  E9 000F5          BLBC    R0, 22$                        ;
                        50     0C   AC  D0 000F8          MOVL    NAMEBLOCK, R0                  ; 1250
                  11    A0          01  88 000FC          BISB2   #1, 17(R0)                     ;
                        50     0C   AC  D0 00100          MOVL    NAMEBLOCK, R0                  ; 1251
                  10    A0          08  88 00104          BISB2   #8, 16(R0)                     ;
                                        04 00108          RET                                    ; 1247
                                    56  D5 00109 22$:     TSTL    COUNT                          ; 1253
```

```
                                        28  15 00010B          BLEQ    26$                                             ;  1256
                                52      02  C0 00010D          ADDL2   #2, BLOCKP                                      ;  1257
                                        5E  DD 00110           PUSHL   SP
                                7E      56  7D 00112           MOVQ    COUNT, -(SP)
                    00000000G   00      03  FB 00115           CALLS   #3, FIL$CVT_DTB
                                05      50  E8 00011C          BLBS    R0, 24$                                         ;
                                    0818 8F  BF 0011F 23$:      CHMU    #2072                                           ;  1258
                                        04  00123              RET                                                     ;
                    00007FFF    8F      6E  D1 00124 24$:      CMPL    VERSION, #32767                                 ;  1259
                                        05  1B 00128          BLEQU   25$                                             ;
                                    0820 8F  BF 0012D          CHMU    #2080                                           ;  1260
                                        04  00131              RET                                                     ;
                                62      6E  B0 00132 25$:      MOVW    VERSION, (BLOCKP)                               ;  1261
                                        04  00135 26$:         RET                                                     ;  1266

; Routine Size:  310 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  279     1267  1 ROUTINE GETCHAR : L_GETCHAR =
;  280     1268  1
;  281     1269  1 !++
;  282     1270  1 !
;  283     1271  1 ! FUNCTIONAL DESCRIPTION:
;  284     1272  1 !
;  285     1273  1 !       This routine returns the RAD-50 code of the next character in the
;  286     1274  1 !       input string if it is in the RAD-50 set. If it is not, or end of
;  287     1275  1 !       string has been reached, it returns zero.
;  288     1276  1 !
;  289     1277  1 ! CALLING SEQUENCE:
;  290     1278  1 !       GETCHAR ()
;  291     1279  1 !
;  292     1280  1 ! INPUT PARAMETERS:
;  293     1281  1 !       NONE
;  294     1282  1 !
;  295     1283  1 ! IMPLICIT INPUTS:
;  296     1284  1 !       COUNT: characters remaining in string
;  297     1285  1 !       STRINGP: string pointer
;  298     1286  1 !       FCOUNT: chars in current field
;  299     1287  1 !
;  300     1288  1 ! OUTPUT PARAMETERS:
;  301     1289  1 !       NONE
;  302     1290  1 !
;  303     1291  1 ! IMPLICIT OUTPUTS:
;  304     1292  1 !       NONE
;  305     1293  1 !
;  306     1294  1 ! ROUTINE VALUE:
;  307     1295  1 !       character code
;  308     1296  1 !
;  309     1297  1 ! SIDE EFFECTS:
;  310     1298  1 !       COUNT decremented and STRINGP advanced if legal character.
;  311     1299  1 !
;  312     1300  1 !--
;  313     1301  1
;  314     1302  2 BEGIN
;  315     1303  2
;  316     1304  2 REGISTER
;  317     1305  2         CHAR            = 5;            ! character in process
;  318     1306  2
;  319     1307  2 EXTERNAL REGISTER
;  320     1308  2         COUNT   = 6,                    ! characters remaining in string
;  321     1309  2         STRINGP = 7     : REF VECTOR [,BYTE],  ! string pointer
;  322     1310  2         FCOUNT  = 8;                    ! count of chars in current field
;  323     1311  2
;  324     1312  2
;  325     1313  2 ! Get the next character from the string and dispatch in its type.
;  326     1314  2 !
;  327     1315  2
;  328     1316  2 CHAR = .STRINGP[0];
;  329     1317  2
;  330     1318  2 CASE TYPE () FROM 0 TO 8 OF
;  331     1319  2     SET
;  332     1320  2     [0,5,6]:                           ! end, dot, or semicolon
;  333     1321  2         CHAR = 0;
;  334     1322  2     [1]:                               ! upper case alpha
;  335     1323  3         BEGIN
```

```
3               CHAR = .CHAR - 'A' + 1;         ! convert to RAD-50 code
3               COUNT = .COUNT - 1;             ! advance to next character
3               STRINGP = .STRINGP + 1;
3               FCOUNT = .FCOUNT + 1;           ! count character in field
2               END;
2           [2]:                                ! lower case alpha
3               BEGIN
3               CHAR = .CHAR - 'a' + 1;         ! convert to RAD-50 code
3               COUNT = .COUNT - 1;             ! advance to next character
3               STRINGP = .STRINGP + 1;
3               FCOUNT = .FCOUNT + 1;           ! count character in field
2               END;
2           [3]:                                ! numeric
3               BEGIN
3               CHAR = .CHAR - '0' + 30;        ! convert to RAD-50 code
3               COUNT = .COUNT - 1;             ! advance to next character
3               STRINGP = .STRINGP + 1;
3               FCOUNT = .FCOUNT + 1;           ! count character in field
2               END;
2           [7]:                                ! dollar sign
3               BEGIN
3               CHAR = 27;                      ! convert to RAD-50 code
3               COUNT = .COUNT - 1;             ! advance to next character
3               STRINGP = .STRINGP + 1;
3               FCOUNT = .FCOUNT + 1;           ! count character in field
2               END;
2           [8]:                                ! underscore
3               BEGIN
3               CHAR = 29;                      ! convert to RAD-50 code
3               COUNT = .COUNT - 1;             ! advance to next character
3               STRINGP = .STRINGP + 1;
3               FCOUNT = .FCOUNT + 1;           ! count character in field
2               END;
2           [4]:                                ! star - legal as only char in field
3               BEGIN
3               CHAR = 0;
3               IF .FCOUNT NEQ 0 THEN ERR_EXIT (SS$_BADFILENAME);
2               END;
2           TES;
2
2   RETURN .CHAR;
2
1   END;                                        ! end of routine GETCHAR


                            55          67  9A 00000 GETCHAR:MOVZBL  (STRINGP), CHAR          ; 1316
                                     0000V  30 00003         BSBW    TYPE                     ; 1318
                    08      00          50  CF 00006         CASEL   R0, #0, #8               ;
      0022        001C    0016        0012     0000A 1$:     .WORD   2$-1$,-                  ;
      0027        0012    0012        0037     00012                 3$-1$,-                  ;
                                      002C     0001A                 4$-1$,-                  ;
                                                                     5$-1$,-                  ;
                                                                     9$-1$,-                  ;
                                                                     2$-1$,-                  ;
```

```
                                                    2$-1$,-                       ;
                                                    6$-1$,-                       ;
                                                    7$-1$                         ;
                     55  D4 0001C 2$:      CLRL     CHAR                          ; 1321
                     2C  11 0001E          BRB      10$                           ;
      55        C0   A5  9E 00020 3$:      MOVAB    -64(R5), CHAR                 ; 1324
                     13  11 00024          BRB      8$                            ; 1325
      55        A0   A5  9E 00026 4$:      MOVAB    -96(R5), CHAR                 ; 1331
                     0D  11 0002A          BRB      8$                            ; 1332
      55             12  C2 0002C 5$:      SUBL2    #18, CHAR                     ; 1338
                     08  11 0002F          BRB      8$                            ; 1339
      55             1B  D0 00031 6$:      MOVL     #27, CHAR                     ; 1345
                     03  11 00034          BRB      8$                            ; 1346
      55             1D  D0 00036 7$:      MOVL     #29, CHAR                     ; 1352
                     56  D7 00039 8$:      DECL     COUNT                         ; 1353
                     57  D6 0003B          INCL     STRINGP                       ; 1354
                     58  D6 0003D          INCL     FCOUNT                        ; 1355
                     0B  11 0003F          BRB      10$                           ; 1318
                     55  D4 00041 9$:      CLRL     CHAR                          ; 1359
                     58  D5 00043          TSTL     FCOUNT                        ; 1360
                     05  13 00045          BEQL     10$                           ;
              0818   8F  BF 00047          CHMU     #2072                         ;
                         05 0004B          RSB                                    ;
      50             55  D0 0004C 10$:     MOVL     CHAR, R0                      ; 1364
                         05 0004F          RSB                                    ; 1366

; Routine Size:  80 bytes,    Routine Base:  $CODE$ + 0136
```

```
;  380     1367  1 ROUTINE GETSTAR : L_GETSTAR =
;  381     1368  1
;  382     1369  1 !++
;  383     1370  1 !
;  384     1371  1 ! FUNCTIONAL DESCRIPTION:
;  385     1372  1 !
;  386     1373  1 !       This routine gobbles the next character in the input string
;  387     1374  1 !       if it is a star.
;  388     1375  1 !
;  389     1376  1 ! CALLING SEQUENCE:
;  390     1377  1 !       GETSTAR ()
;  391     1378  1 !
;  392     1379  1 ! INPUT PARAMETERS:
;  393     1380  1 !       NONE
;  394     1381  1 !
;  395     1382  1 ! IMPLICIT INPUTS:
;  396     1383  1 !       COUNT: number of characters in input string
;  397     1384  1 !       STRINGP: input string pointer
;  398     1385  1 !
;  399     1386  1 ! OUTPUT PARAMETERS:
;  400     1387  1 !       NONE
;  401     1388  1 !
;  402     1389  1 ! IMPLICIT OUTPUTS:
;  403     1390  1 !       NONE
;  404     1391  1 !
;  405     1392  1 ! ROUTINE VALUE:
;  406     1393  1 !       1 if character was a star
;  407     1394  1 !       0 otherwise
;  408     1395  1 !
;  409     1396  1 ! SIDE EFFECTS:
;  410     1397  1 !       COUNT decremented, STRINGP incremented if character was star.
;  411     1398  1 !
;  412     1399  1 !--
;  413     1400  1
;  414     1401  2 BEGIN
;  415     1402  2
;  416     1403  2 EXTERNAL REGISTER
;  417     1404  2         COUNT    = 6,                   ! characters remaining in string
;  418     1405  2         STRINGP = 7     : REF VECTOR [,BYTE];  ! string pointer
;  419     1406  2
;  420     1407  2 IF .COUNT GTR 0 AND .STRINGP[0] EQL '*'
;  421     1408  2 THEN
;  422     1409  3     BEGIN
;  423     1410  3     COUNT = .COUNT - 1;
;  424     1411  3     STRINGP = .STRINGP + 1;
;  425     1412  3     1
;  426     1413  3     END
;  427     1414  2 ELSE
;  428     1415  2     0
;  429     1416  2
;  430     1417  1 END;                                   ! end of routine GETSTAR


                                56  D5 00000 GETSTAR:TSTL    COUNT                                ; 1407
```

```
                          0D  15 00002          BLEQ    1$                                      ;
                    2A    67  91 00004          CMPB    (STRINGP), #42                          ;
                          08  12 00007          BNEQ    1$                                      ;
                          56  D7 00009          DECL    COUNT                                   ; 1410
                          57  D6 0000B          INCL    STRINGP                                 ; 1411
                    50    01  D0 0000D          MOVL    #1, R0                                  ; 1409
                                05 00010          RSB                                             ;
                          50  D4 00011 1$:      CLRL    R0                                      ; 1407
                                05 00013          RSB                                             ; 1417

; Routine Size:  20 bytes,    Routine Base:  $CODE$ + 0186
```

```
;  432     1418  1 ROUTINE TYPE : L_TYPE =
;  433     1419  1
;  434     1420  1 !++
;  435     1421  1 !
;  436     1422  1 ! FUNCTIONAL DESCRIPTION:
;  437     1423  1 !
;  438     1424  1 !       This routine determines the type code of the current character
;  439     1425  1 !       in the string.
;  440     1426  1 !
;  441     1427  1 ! CALLING SEQUENCE:
;  442     1428  1 !       TYPE ()
;  443     1429  1 !
;  444     1430  1 ! INPUT PARAMETERS:
;  445     1431  1 !       NONE
;  446     1432  1 !
;  447     1433  1 ! IMPLICIT INPUTS:
;  448     1434  1 !       COUNT: number of characters left in string
;  449     1435  1 !       STRINGP: string pointer
;  450     1436  1 !
;  451     1437  1 ! OUTPUT PARAMETERS:
;  452     1438  1 !       NONE
;  453     1439  1 !
;  454     1440  1 ! IMPLICIT OUTPUTS:
;  455     1441  1 !       NONE
;  456     1442  1 !
;  457     1443  1 ! ROUTINE VALUE:
;  458     1444  1 !       type code of character:
;  459     1445  1 !       0: end of string or non-RAD-50
;  460     1446  1 !       1: upper case alpha
;  461     1447  1 !       2: lower case alpha
;  462     1448  1 !       3: numeric
;  463     1449  1 !       4: star
;  464     1450  1 !       5: dot
;  465     1451  1 !       6: semicolon
;  466     1452  1 !       7: $
;  467     1453  1 !       8: _
;  468     1454  1 !
;  469     1455  1 ! SIDE EFFECTS:
;  470     1456  1 !       NONE
;  471     1457  1 !
;  472     1458  1 !--
;  473     1459  1
;  474     1460  2 BEGIN
;  475     1461  2
;  476     1462  2 EXTERNAL REGISTER
;  477     1463  2         COUNT   = 6,            ! characters remaining in string
;  478     1464  2         STRINGP = 7     : REF VECTOR [,BYTE]; ! string pointer
;  479     1465  2
;  480     1466  2 ! Character match tables. First is low character of range, second is
;  481     1467  2 ! high character. Type is table index of the matching range.
;  482     1468  2 !
;  483     1469  2
;  484     1470  2 BIND
;  485     1471  2         LOWCHAR         = UPLIT BYTE (0, 'Aa0*.;$_') : VECTOR [,BYTE],
;  486     1472  2         HIGHCHAR        = UPLIT BYTE (0, 'Zz9* ;$_') : VECTOR [,BYTE];
;  487     1473  2
;  488     1474  2 ! If the string is empty return 0 as the type. Else search the tables.
```

```
;  489          1475  2 !
;  490          1476  2
;  491          1477  2 IF .COUNT LEQ 0 THEN RETURN 0;
;  492          1478  2
;  493          1479  2 INCR I FROM 1 TO 8 DO
;  494          1480  2     IF .STRINGP[0] GEQU .LOWCHAR[.I]
;  495          1481  2     AND .STRINGP[0] LEQU .HIGHCHAR[.I]
;  496          1482  2     THEN RETURN .I;
;  497          1483  2
;  498          1484  2 ERR_EXIT (SS$_BADFILENAME);              ! other characters are illegal
;  499          1485  2
;  500          1486  2
;  501          1487  1 END;                                     ! end of routine TYPE


                                                00  0019A P.AAA:   .BYTE   0                                       ;
                5F  24  3B  2E  2A  30  61      41  0019B          .ASCII  \Aa0*.;$_\                              ;
                                                00  001A3 P.AAB:   .BYTE   0                                       ;
                5F  24  3B  2E  2A  39  7A  5A  001A4              .ASCII  \Zz9*.;$_\                              ;

                                                        LOWCHAR=            P.AAA
                                                        HIGHCHAR=           P.AAB


                                            56  D5 00000 TYPE:    TSTL    COUNT                                   ; 1477
                                            1A  15 00002          BLEQ    3$                                      ;
                          50                01  D0 00004          MOVL    #1, I                                   ; 1480
                    E2 AF40                 67  91 00007 1$:      CMPB    (STRINGP), LOWCHAR[I]                   ;
                                            07  1F 0000C          BLSSU   2$                                      ;
                    E4 AF40                 67  91 0000E          CMPB    (STRINGP), HIGHCHAR[I]                  ; 1481
                                            08  1B 00013          BLEQU   4$                                      ;
                EE            50            08  F3 00015 2$:     AOBLEQ  #8, I, 1$                               ; 1480
                                    0818    8F  BF 00019          CHMU    #2072                                   ; 1484
                                            05  0001D          RSB                                             ;
                                            50  D4 0001E 3$:     CLRL    R0                                      ; 1487
                                            05  00020 4$:     RSB                                             ;

; Routine Size:  33 bytes,    Routine Base:  $CODE$ + 01AC

;  502          1488  1
;  503          1489  1 END
;  504          1490  0 ELUDOM



;                                       PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $CODE$                           461  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                              Library Statistics
;
;                                      --------- Symbols ---------      Pages       Processing
;       File                           Total     Loaded    Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1       18619        25          0       1000          00:02.0
;
;
;
;
;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:MAKNMB/OBJ=OBJ$:MAKNMB MSRC$:MAKNMB/UPDATE=(ENH$:MAKNMB)
;
; Size:          443 code + 18 data bytes
; Run Time:          00:15.7
; Elapsed Time:      00:33.1
; Lines/CPU Min:      5694
; Lexemes/CPU-Min: 21183
; Memory Used:  176 pages
; Compilation Complete
```

PMS
LIS

QUOTAUTIL
LIS

MAKPTR
LIS

MATCHNAME
LIS

MPWIND
LIS

LOCKDB
LIS

LOCKERS
LIS

MAKACC
LIS

PARSNM
LIS

MAPVBN
LIS

IODONE
LIS

LOCKDN
LIS

MODIFY
LIS

MOUNT
LIS

NXTHDR
LIS

MAKNMB
LIS

MAKSTR
LIS